Sensor Webs as Multiagent, Negotiating Systems

Costas Tsatsoulis
Information and Telecommunication Technology Center (ITTC)
Department of Electrical Engineering and Computer Science
The University of Kansas
Lawrence, KS 66045
tsatsoul@ittc.ku.edu

Abstract—We describe how we developed a multi-agent system to represent a collection of sensors that are dynamically combined into a Sensor Web. The sensors use a combination of a Matchmaker architecture together with the Contract Net protocol to enable the reasoned, task-based creation of a dynamic Sensor Web that adapts to the data sensed and to the requirements generated by the agent sensors. We tested our system using EO-1 instruments, and showed how Sensor Web instrument coalitions can be generated based on sensing and processing needs.

1. INTRODUCTION

A Sensor Web is a dynamically formed, distributed system consisting of individual sensing nodes which can be orbital or terrestrial, *in situ* or remote, sessile or mobile, and which have the ability to communicate and exchange information. Sensor Webs have been demonstrated successfully in variety of environmental sensing applications, including oxygen gas sensing [1] and soil moisture sensing [2]. These demonstrations have also shown how the collected information can control external systems, such as sprinklers; how it can handle spatio-temporal sensing; and how it can increase the robustness of the sensing network.

Our work takes Sensor Webs into a new phase, by making them *intelligent*, *collaborative*, and *self-aware*. The intelligence is added to each sensor, and allows the Sensor Web to be truly adaptive to the events in the environment it senses. The Sensor Web becomes a multi-agent system, where individual *sensor agents* can collaborate, form dynamic coalitions to handle tasks that a single sensor cannot, and can negotiate in order to form the best set of coalitions that maximize the overall utlility of the Sensor Web.

Each sensor identifies tasks that it cannot perform, and then uses the Contract Net protocol to ask other agents to assist it. We applied our approach to some of the Earth Observing 1 (EO-1) instruments, and have shown that the agents can correctly form a Sensor Web that will satisfy the dynamically generated sensing and processing needs.

2. SENSOR AGENT ARCHITECTURE

Each sensor hosts an intelligent agent that adds to the sensor autonomous decision making. The sensor agent[1] consists of two basic components as shown in Figure 1, the *Reasoning Component* and the *Communication Component*. The Communication component is responsible for receiving messages from and delivering messages to the other sensor agents in the Sensor Web. The Reasoning component consists of four sub-components. The *event monitor* snoops the data collected by the sensor, and, if it identifies interesting events, it informs the Reasoning component. While the event monitor reacts to events, the *task generator* uses these events to predict future sensing requirements, needs, and parameters. The Reasoning component decides whether the sensor can satisfy on its own the new sensing needs arising from the event (by checking its own state represented by the *agent state* sub-component), or whether it will require the assistance of other sensors. In the latter case, the agent generates *task requests* through its task component, and contacts other agents that it believes can assist it. These other agents will choose to respond to this request, and if they respond, they will do so through a *bid.* The *bid evaluator* sub-component examines the bid, and, if satisfactory, it awards the task to some other sensor agent.

After the task request, bidding, bid evaluation, and task assignment phases, the coalition of sensors is finalized, and then the newly created Sensor Web proceeds with its sensing tasks. Since the task scheduler allows the Sensor Web to schedule future tasks, the sensing coalition may change over time, since different sensors may exit and enter it, as the sensing tasks and requirements change over time.

[1] In this paper we will use "agent" and "sensor agent" interchangeably to indicate the intelligent agent associated with a specific sensor or sensors.

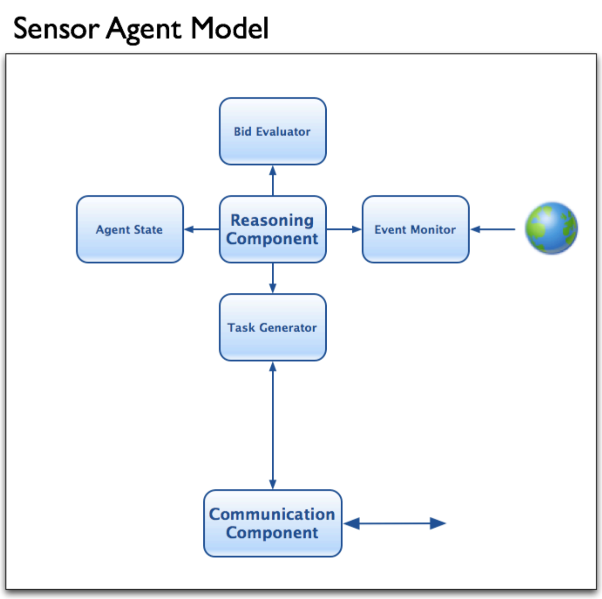

Figure 1: Architecture of a sensor agent

3. SENSOR AGENT OPERATIONS

The operations of our sensor agents are summarized in Figure 2: A sensor is initialized by its sensor description, represented in SensorML [3]. SensorML is a standard by the Open Geospatial Consortium that provides a description of sensor systems. This description allows an agent to be aware of is own capabilities. As the sensor senses the environment, its event monitors may identify an event or set of data that necessitate additional sensing and processing tasks that cannot be performed by the sensor. The sensor agent will attempt to identify the sensors that *can* perform these tasks by contacting a *Matchmaker*. The Matchmaker provides the sensor agent with the names and contacts of the sensors it believes can perform the necessary tasks. Next, the sensor agent composes bid requests for these sensor agents, and asks them to bid for the tasks. When bids are submitted, it evaluates them, and then it awards the sensing or processing contract to the best suited sensor. The process of contract award leads to the formation of a coalition of sensors that can collaborate to perform complex sensing tasks, i.e. it leads to the formation of a Sensor Web.

All agent communicative acts follow the Foundation for Intelligent Physical Agents (FIPA) and Resource Description Framework (RDF) standards [4, 5]. FIPA is an IEEE Computer Society standards organization that "promotes agent-based technology and the interoperability of its standards with other technologies." [4]. FIPA defines, among others, the message parameters for multi-agent systems, including the contents and semantics of a message, and a variety of message protocols. RDF was created in 1999 as a standard on top of the Extensible Markup Language (XML) for encoding metadata for the semantic web [6]. The 2004 updated RDF standard moved beyond that and allows the encoding of ontological information about things in the world and their relationships between them [5]. The RDF standard is supported by the World Wide Web Consortium (W3C) [7].

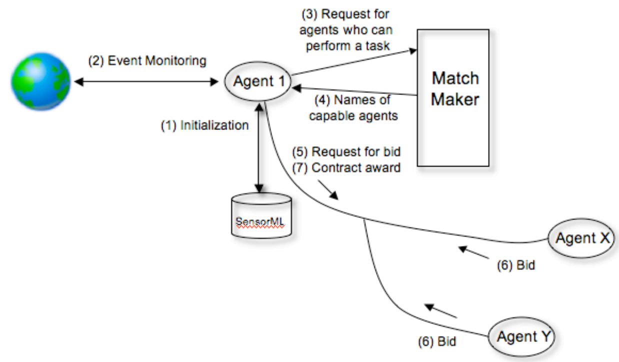

Figure 2: Operations of a sensor agent

Sensor Representation

Sensors in our work are represented using a subset of SensorML [3]. In our work we chose to represent some of the instruments on EO-1. Specifically, we represent the Advance Land Imager (ALI), Hyperion, and LEISA/Atmospheric Corrector (LAC). The sensor representation is simpler than one that is possible through a complete SensorML formalism (actually, some of these instruments have no complete SensorML description yet), and our goal was to represent the basic sensor capabilities so as to allow reasoning. For example, the Advanced Land Imager (ALI) sensor of EO-1 is a multispectral imager with a spectral range from 0.4 to 2.4 µm. ALI covers from the visible to the near infrared to the short-wave infrared spectrum, and has a resolution of 30 m except for its panchromatic band that has a resolution of 10 m. Our representation refers to ALI's sensing capabilities in terms of the bands, resolution, signal to noise ration, etc., plus capabilities of the vehicle, such as on-board processing and storage.

Event Monitoring and Generation of Tasks

A sensor operates in some standard mode, sensing its environment, processing the data, and sharing some parts of it with users and other sensors. At the same time, the sensor agent uses *event monitors* to identify events of interest. Events of interest are driven by the satisfaction of a logical condition involving state data[2] (*data-driven events*). Data-driven events can be *instantaneous* and conditioned on the current values of state variables that reflect the current state of the world, or *continuous*, for example trends or averages involving those variables. Continuous data-driven events require storage of previous value(s) in some form. For

[2] "State data" are any data that are associated with the agent or the data it is collecting through its sensor(s).

example, an event condition that checks on a running average or a trend would be classified as continuous, since prior state information would have to be stored.

The event monitor is developed as a collection of active, conditional elements that match against the state variables of the sensor. For example, an instantaneous event monitoring rule would be:

```
if (?SOIL-TEMP > 100°F) then ACTIVATE-AGENT
```

indicating that if the sensed parameter "soil temperature" is above 100 degrees Fahrenheit, the sensor agent should be activated. Or:

```
if (?FREE-DISC-SPACE < 1000) then ACTIVATE-
AGENT
```

indicating that if the sensor parameter "free disc space" is less than 1000 units (bytes, Kbytes, etc.), the sensor agent should be activated. Note that state variables can be both internal to the sensor (i.e. its own state, tasks, status) and external to it (e.g. sensed data, requests by other sensors)

A continuous event monitoring rule would be:

```
if (?SOIL-TEMP increased by 50% in 30
minutes) then ACTIVATE-SPRINKLER
```

Such a rule requires that the event monitor store historical soil temperatures for at least the last 30 minutes.

The interesting event data are also shared with the *task generator*. The task generator identifies which tasks cannot be satisfied by the sensor, and generates task requests.

The Matchmaker

The Matchmaker is a centralized repository of sensor agent capabilities and needs [8]. When sensor agents enter the multi-agent Sensor Web environment, they advertise their sensing and processing capabilities to the Matchmaker. An advertisement example is given below (to allow for readability we have stripped the FIPA and RDF tags, and we are showing only the messages our program will print out):

```
ALI_Pan: Registering provided services.
Matchmaker:Agent ALI_Pan registered:
[urn:ogc:def:classifier:Spectrum] Visible
[urn:ogc:def:classifier:intendedApplication]
CloudSensing
[urn:ogc:def:classifier:intendedApplication]
LandSensing
```

This Matchmaker capabilities registration message comes from the ALI Panchromatic sensor, and indicates that it can sense in the visible spectrum, and can sense clouds and land.

Capabilities registered with a Matchmaker are always described on a high level. There are two reasons for that: First, detailed capabilities might change over time (for example, the area sensed by a mobile sensor such as one on a space vehicle). Second, an agent requesting a capability may be willing to settle for something worse than what it asked for (for example, it may like imagery with 100 m. resolution, but anything under 1 km. resolution might be acceptable compared to no data). If all capabilities were advertised in perfect detail, it would become impossible to make compromises between requested and available sensor services.

In addition to capabilities, agents may register requests with the Matchmaker. A request indicates that the requesting agent is looking for others that can satisfy its needs by performing the required tasks. The following is an example of a registered need:

```
Manager: Requesting agents with my required
services:
[urn:ogc:def:classifier:sensorSpectrum] SWIR
[urn:ogc:def:classifier:intendedApplication]
LandSensing
```

This message asks for help from agent(s) who can perform sensing of land using a short-wave infrared sensor.

When the Matchmaker receives a request for capabilities, it looks at the capabilities that agents have registered with it, and sends the list of capable agents to the requesting agent.

Contracting for Sensing and Processing Tasks

Our work is based on the seminal research on Contract Nets described in [9] and [10]. In the Contract Net protocol, agents request that others submit bids for completing specific tasks, and then evaluate these bids to identify the most capable agent. Our work adopts this approach and adapts it to the environment of Sensor Webs.

After one of our sensor agents has generated a task and has received from the Matchmaker a list of agents that could perform it, it contacts these agents directly with a *request for bids*. Each bid request consists of a *task abstraction* and a *bid specification*. The task abstraction is a description of the task in finer level of detail than the one sent to the Matchmaker, but still not completely specified, since the requesting agent may need to make tradeoffs when it receives the task bids. What the task abstraction indicates to the agents receiving the bid request is that if they cannot satisfy *at least* that requirement, they should not bid.

The bid specification indicates the contents of the response to the bid. These are the items the bid requesting agent (*manager*) will use to evaluate the submitted bid. The following is an example of a bid request:

```
Manager:Sending out bid requests: task
abstraction:
(Integer) wavelength='1200'
bid specification:
[signal-To-NoiseRatio, resolution]
```

This example bid request indicates that the bid submitting agent *must* be able to sense at 1200 nanometer wavelength. If an agent is going to submit a bit, it must indicate its signal to noise ratio and its resolution at that frequency, which means that the bid requester will use these two parameters to evaluate bids before awarding the task to another sensor agent.

Sensor agents receiving a bid request decide if they can perform the task (by looking at the task abstraction), and if their requested sensing or processing capabilities are not already allocated. If they can respond to a bid request they do so by submitting a bid as the one below, which was sent from the Hyperion sensor agent of EO-1:

```
Hyperion: Instrument meets task abstraction.
Sending bid:
(Quality) signalToNoiseRatio='GOOD'
(Double) resolution='30.0'
```

The bid requesting agent receives bids and waits a (currently) predefined period of time before it starts evaluating them (the waiting time is used in lieu of requiring agents to respond with a "will not submit a bid" message). The evaluation is based on the bid specification, and the bid submitting agent that meets the required specifications the closest is assigned the task contract.

The use of a Contract Net protocol for sensor task assignment achieves a number of goals: First, only sensor agents that can perform the task respond to a bid request, limiting the communication overhead of the system. Second, by advertising general needs, the requesting agent can use internal, domain-specific evaluation criteria to assign the task to a sub-optimal agent, but to one who can still provide a satisfactory solution; this allows us to deal with over-constrained task requests that have no optimal solution. Third, the approach makes optimal use of the sensor network, by exchanging the minimum possible number of messages, and by assigning a task to the most capable and available sensor. Finally, the approach responds to a changing environment that requires a dynamic allocation of tasks, and which leads to the formation of an adaptive, intelligent Sensor Web.

4. RELATED WORK

Some work has started to add to Sensor Webs the ability to adapt and react to the environment. For example, Jain *et al.* [11] discuss how the DeSiDeRaTa adaptive resource management strategy could be used by constellations of satellites to dynamically reconfigure their computational load. Some other work has looked at changing the configuration of a sensor web using mobile sensors to improve sensing a thermocline [12]. The work by Chien *et al.* with the EO-1 system showed how to dynamically configure its sensors based on user requests [13,14]. There is also a lot of work making proposals of how Sensor Webs could be best used, such as the proposal described in [15].

Our work differs from these and many other similar efforts. We implement a general, intelligent, adaptive, and testable mechanism to form coalitions of pods that can perform sensing tasks. Our approach is not specific to some particular sensing task or sensor type. Our work is applicable to all Sensor Web tasks, since it develops the overall Sensor Web formation methodology, and individual components such as the specific event monitors or particular sensor capabilities can be changed for different types of Sensor Webs and sensors.

Our work touches upon the areas of event monitoring. Active database systems provide event "alerters" (triggers) that notify applications of interesting data events [16]. However, an alert mechanism can only let an agent know about relevant changes, but it does not support an agent whose information needs change due to changes in an environment, as our proposed event monitoring.

There has been some previous work in using multi-agent systems to control suites of sensors. For example, Soh *et al.* describe forming coalitions of sensors to track moving vehicles in real time [17,18]. Modi *et al.* approach a similar sensor coalition problem from the perspective of distributed resource allocation [19], while Horling *et al.* use the Task Analysis, Environmental Modeling and Simulation (TAEMS) language to quantitatively describe alternate ways a goal can be achieved, and, consequently, allow alternate ways of assembling sensors in a coalition [20,21]. This and similar work in sensor networks deals with sessile, homogeneous sensors, which are dedicated to a single sensing task (e.g. tracking vehicles), and where the goal of the multi-agent architecture is to pick the best sensors for the task. Our work in Sensor Webs deals with heterogeneous sensors which can be mobile, and which must come together into a dynamic coalition to satisfy an *a priori* unkown, dynamically formulated task.

There is work in multi-agent reconnaissance (for example, see [22], [23], and [24]), but this work focuses on task planning, task coordination, and task precedence, and not on forming sensing coalitions of heterogeneous sensing assets. Finally, there are some proposals for multi-agent reconnaissance for planetary exploration, but no concrete system has been deployed [25].

5. CONCLUSIONS

In this paper we describe how a combination of a Matchmaker and Contract Net architecture allows for the dynamic formation of Sensor Webs. In our approach sensors use SensorML to represent their abilities, and then register these with a central capability repository, the Matchmaker. The sensor agents actively monitor the environment, and when it is so required, they post task needs with the Matchmaker. These needs indicate tasks that they cannot complete on their own, and for which they need the generation of a Sensor Web. After the Matchmaker provides them with the agents who could satisfy these needs, the agents use the Contract Net protocol to request and evaluate bids for these tasks. Under this protocol agents submit bids for the completion of tasks, and the bid requesting agent uses domain-specific criteria to evaluate these bids and then award task contracts, thus recruiting agents to the Sensor Web. In our work we used the EO-1 sensing instrument (ALI, Hyperion, and LAC) as potential sensors, and have shown how the overall approach allows for the formation of a Sensor Web.

Future work will expand the Contract Net protocol by allowing negotiation for the use of limited constrained resources, and by allowing sensors to share their resources among many requests by agents. We will also compare our approach to the formation of a Sensor Web with other competing ones, such as ones based on a Service Oriented Architecture [26].

REFERENCES

[1] Delin, K., and S.P. Jackson. 2000. "Sensor Web for *in situ* Exploration of Gaseous Biosignatures," *Proc. IEEE Aerospace Conference,* Big Sky, MT, 465-472.

[2] Teillet, P. R. Gauthier, G. Fedosejevs, M. Maloley, A. Chichagov, and G. Ainsley. 2003. "A Soil Moisture Monitoring Sensorweb Demonstration in the Context of Integrated Earth Sensing," *Proc. of SPIE*, vol. 5151, Earth Observing Systems VIII, William L. Barnes (Ed.), 63-73.

[3] Botts, M. (ed.). 2002. *Sensor Model Language (SensorML) for In-site and Remote Sensors,* OpenGIS Interoperability Program Report, OGC 02-026, v. 07, Open GIS Consortium Inc.

[4] FIPA. 2007. "Foundation for Intelligent Physical Agents," www.fipa.org (last accessed October 20, 2007).

[5] Lassila, O. and R. Swick. (eds.). 1999. *Resource Description Framework (RDF) Model and Syntax Specification.* Recommendation, W3C.

[6] W3C. 2007. www.w3.org. (last accessed October 21, 2007).

[7] XML. 2007. www.xml.org. (last accessed October 21, 2007).

[8] Kuokka, D. and L. Harada. 1996. "Integrating Information via Matchmaking," *J. of Intelligent Information Systems,* vol. 6, no. 2-3, 261-279.

[9] Smith, R. 1980. "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Trans. on Computers,* vol. C-29, no. 12, 1104-1113.

[10] Smith, R. and R. Davis. 1981. "Frameworks for Cooperation in Distributed Problem Solving," *IEEE Trans. on Systems, Man, and Cybernetics,* vol. SMC-11, no. 11, 61-70.

[11] Jain, S., L. Welch, D. Chelberg, Z. Tan, D. Fleeman, D. Parrott, D. Pfarr, M. Liu, and C. Shuler. 2002. "Collaborative Problem Solving Agent for On-Board Real-Time Systems," *Int. Parallel and Distributed Processing Symposium.*

[12] Zhang, B., G. Suhkatme and A. Requicha. 2004. "Adaptive Sampling for Marine Microorganism Monitoring," *IEEE/RSJ Int. Conf. on Intelligent Robotics and Automation.*

[13] Chien, S., R. Sherwood, D. Tran, B. Cichy, G. Rabideau, *et al.* 2005. "The Autonomous Sciencecraft Embedded Systems Architecture," *IEEE Int. Conf. on Systems, Man and Cybernetics*, Hawaii, USA.

[14] Chien, S., B. Cichy, A. Davies, D. Tran, G. Rabideau, *et al.* 2005b. "An Autonomous Earth Observing Sensorweb," *IEEE Int. Conf. on Systems, Man and Cybernetics,* Hawaii, USA.

[15] Koratkar, A., J. Grosvenor, J. Jung, and J. Geiger. 2005. "Autonomous Multi-sensor Coordination: The Science Goal Monitor," *Proc. of SPIE*, vol. 5659, 293-300.

[16] Hanson, E.N., and Noronha, L.X. 1999. "Timer-driven Database Triggers and Alerters: Semantics and a Challenge," *SIGMOD Record*, vol. 28 (4), 11–16.

[17] Soh, L-K., C. Tsatsoulis, and H. Sevay. 2003. "A Satisficing, Negotiated, and Learning Coalition Formation Architecture," in: *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz, Jr. and M. Tambe (Eds.), Boston: Kluwer Academic Publishers, 109-138.

[18] Soh, L-K. and C. Tsatsoulis. 2005. "A Real-Time Negotiation Model and a Multi-Agent Sensor Network Implementation," *Autonomous Agents and Multi-Agent Systems,* vol. 11, no. 3, 215-271.

[19] Modi, P.J., P. Scerri, W-M. Shen, and M. Tambe. 2003. "Distributed Resource Allocation," in: *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz, Jr. and M. Tambe (Eds.), Boston: Kluwer Academic Publishers, 219-256.

[20] Horling, B., R. Mailler, J. Shen, R. Vincent, and V. Lesser. 2003. "Using Autonomy, Organizational Design and Negotiation in a DSN," in: *Distributed Sensor Networks: A Multiagent Perspective,* V. Lesser, C.L. Ortiz, Jr. and M. Tambe (Eds.), Boston: Kluwer Academic Publishers, 139-184.

[21] Horling, B., R. Vincent, R. Mailler, , J. Shen, R. Becker, K. Rawlins, and V. Lesser. 2001. "Distributed Sensor Network for Real Time Tracking," *Proc. of the 5th Int. Conf. on Autonomous Agents,* Montreal, Canada, ACM Press, 417-424.

[22] Giovanini, L., J. Balderud, and R. Katebi. 2007. "Autonomous and Decentralized Mission Planning for Clusters of UUVs," *Int. J. of Control,* 80:7, 1169-1179.

[23] Chandler, P., M. Pachter, S. Rasmussen, and C. Schumacher. 2002. "Multiple Task Assignment fo a UAV Team," *AIAA Guidance, Navigation, and Control Conference,* Anchorage, Alaska, USA, 4587-1836.

[24] Kang, W., A. Sparks, and S. Banda. 2001. "Coordinated Control of Multi-Satellite Systems," *J. of Guidance, Control, and Dynamics,* 24, 360-368.

[25] Fink, W., J.M. Dohm, M.A. Tarbell, T.M. Hare, and

V.R. Baker. 2005. "Next-Generation Robotic Planetary Reconnaissance Missions: A Paradigm Shift," *Planetary and Space Science,* vol. 53, iss. 14-15, 1419-1426.

[26] Erl, T. 2007. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall.